

692

81-070A-06C
EICS SOFTWARE TAPE

81-070A-06D
EICS SATF FILES-FLUXES; DENSITIES DISK

Dynamics Explorer 1

Energetic Ion Composition Spectrometer

(EICS) Software Tape
81-070A-06C

Stand Alone Telemetry Format (SATF) Files-Fluxes, Densities-Disk
81-070A-06D

This data set, 81-070A-06C, consists of one magnetic tape. It is 1600 BPI, ASCII, written in Files 11 format on the VAX computer using the copy command. It is self documenting. The D and C numbers follow:

D-82238

C-27940

LABEL NAME = EICS

The data set, 81-070A-06D, consists of five optical disks and four 8mm tapes. The disks are written in ASCII Files 11 format on the VAX 8650 computer. The tapes are low density, written in VAX Backup format, when restored to disk are organized in the same manner as the original Optimem optical disks. All documentation, software and formats are on the disks and tapes. The file name indicates the year and day of the data contained within that file. The times are not in consecutive sequence. Refer to the latest "Eics Data Listing" to determine exactly which disk has desired data. The volume label name, disk number, and time spans follow:

DISK #	C #	LABEL NAME	TIME SPAN
KV00002		DEA6_0001A	09/15/81 - 09/13/84
KV00007		DEA6_0002A DEA6_0002B	09/15/81 - 06/02/85 09/16/81 - 07/07/86
KV00008		DEA6_0003A DEA6_0003B	09/16/81 - 06/02/87 10/08/81 - 10/09/87
KV00015		DEA6_0004A DEA6_0004B	09/24/81 - 03/24/88 09/20/81 - 04/18/88
KV00016		DEA6_0005A DEA6_0005B	02/01/82 - 05/27/89 06/28/82 - 05/04/90
DD87900	C-031022	EICS6A	09/15/81 - 07/14/90
DD87901	C-031023	EICS6B	01/21/82 - 10/04/91
DD87902	C-031024	EICS7A	10/15/82 - 10/29/90
DD87903	C-031025	EICS7B	01/01/82 - 10/06/90

AAA README. FIRST

1082258

This file name is AAAREADME.FIRST It includes a directory of all of the files on this tape.

The tape was created in September 1990 by W.K. Peterson

Lockheed Space Science Laboratory
SPAN LOCKHD::PETE
415-424-3269

It contains

source code,	(*.for, *.pas, *.mar)
'INCLUDED' fortran files	(*.cmn, *.inc)
pascal environment files	(*.pen)
object modules,	(*.obj)
listings of the compiled modules	(*.lis)
procedures to compile and link	(*.com)
map of executable image	(EICS.MAP)
executable image	(EICS.EXE)

for the program EICS that reads Stand Alone Telemetry Files for the Energetic Ion Composition Spectrometer (EICS) on the Dynamics Explorer -1 satellite. The code was compiled and linked on a LAVC consisting of an 11/780 boot node

Also contained on this tape is a file called directory.lis which is a directory of the files on this tape.

The following warning messages are given when the modules CIAOS.pas, INSEC.pas, IOSUBS.pas, and KMISC.pas are compiled.

CIAOS.pas

%PASCAL-W-ENVWARN, (1) Environment resulted from a compilation with Warnings
%PASCAL-W-ENVWARN, (2) Environment resulted from a compilation with Warnings
%PASCAL-W-UNWRITTEN, (1) Variable RSTRING is read, but never assigned into

INSEC.pas

%PASCAL-W-ENVWARN, (1) Environment resulted from a compilation with Warnings
%PASCAL-W-ENVWARN, (2) Environment resulted from a compilation with Warnings

IOSUBS.pas

%PASCAL-W-ENVWARN, (1) Environment resulted from a compilation with Warnings
%PASCAL-W-NOTVOLATILE, (1) INPUT is non-VOLATILE
%PASCAL-W-NOTVOLATILE, (1) PASCAL_FILE is non-VOLATILE
%PASCAL-W-ADDRESSVAR, (1) PASCAL_FILE is a VAR parameter, ADDRESS is illegal

KMISC.pas

%PASCAL-W-ACTPASNVTMP, (1) Conversion: actual passed is resulting temporary

The ENVIRONMENTAL warnings in the CIAOS.pas and INSEC.pas and

IOSUBS.pas compilations come from slightly non-standard Pascal usage in the routine KMISC.pas. The code in KMISC.PAS that generates the warning and the compilation warning is as follows:

```
-----
1 0 [global] procedure filespecp(filename: varying[11] of char;
1 0           var tranname: varying[12] of char);
1 0   var
1 0     ftranname: packed array[1..255] of char;
1 0     tranlen: integer;
1 1   begin
1 1     filespec(filename, ftranname, tranlen);
1           1
%PASCAL-W-ACTPASCNVTMP, (1) Conversion: actual passed is resulting temporary
1 1     tranname := substr(ftranname, 1, tranlen);
0 0   end;
-----

```

This code is passing a variable string from pascal to Fortran but converting into a Fortran string descriptor. The warning says that if the variable is changed by further Fortran processing the Pascal variable is not changed. In this application that is exactly what we want. kvs 8/89

In addition to an Environmental warning noted above, compilation of the module IOSUBS.PAS generates warnings related to the non standard use of a pointer. Code comments relating to the two sections that contain the subject code follow:

```
-----  
1 0 [global] procedure get_linep (instr: varying[len] of char;  
1 0           var rstring: varying[12] of char;  
1 0           flageof, flagerror: Boolean := false;  
1 0           return_null: Boolean := false);  
C 1 0 {  
C 1 0   Written March, 1986 by K. Van Stone  
C 1 0   Latest version: 2-December-1987  
C 1 0  
C 1 0   Modified April 1986 by KVS to trap errors and eof depending on the  
C 1 0       value of flageof and flagerror.  
C 1 0   Modified June, 1986 so that it can be called from outside.  
C 1 0   Modified August, 1986 so an end-of-file on input won't result  
C 1 0       in an infinite loop.  
C 1 0   Modified April, 1987 to call quit on end of file.  
C 1 0   Modified December, 1987 to use various input types (see  
C 1 0       set_input_type).  
C 1 0  
C 1 0   Writes instr and gets the response. Won't accept null responses  
C 1 0       unless return_null is set to true.  
C 1 0  
C 1 0   Input variables:  
C 1 0     instr: string      - Prompt to the user. Prompt is ignored when  
C 1 0               current_input.input_type = file_io and  
C 1 0               sometimes when = fortran_io.  
C 1 0     flageof: boolean    - = true if the routine should return upon  
C 1 0               EOF, storing the value in eof_status, =  
C 1 0               false if the routine should abort on EOF,  
C 1 0               Default is false.  
C 1 0     flagerror: boolean  - = true if the routine should trap read  
C 1 0               errors and store them in error_status, =  
C 1 0               false if the routine should abort instead.  
C 1 0               Default is false.  
C 1 0     return_null: boolean - = true if the routine should return when it  
C 1 0               encounters an empty response (rstring = ''),  
C 1 0               = false if the routine should prompt again  
C 1 0               until the response is non-null. Default is  
C 1 0               false.  
C 1 0   Output variable:  
C 1 0     rstring: string      - The user response.  
C 1 0  
C 1 0   Also uses the global variable current_input to see how to read the  
C 1 0       string.  
C 1 0 }  
1 0   var  
1 0     pascal_file: ^text;
```

```

1 0      rbuff: packed array [1..512] of char;
1 0      rlen: integer;
1 1 begin
1 1     eof_status := false;
1 1     error_status := 0;
1 2 repeat
1 3     case current_input.input_type of
1 3 terminal_io, file_io:
1 4 begin
1 4     if (current_input.input_type = terminal_io) then
1 5 begin
1 5     write(instr);
1 5     pascal_file := address(input);
1
%PASCAL-W-NOTVOLATILE, (1) INPUT is non-VOLATILE
%PASCAL-I-NOTBEADDR, (1)           - may not be parameter to ADDRESS

```

This code sets a pointer (pascal_file) that points to the input source. In this case we are using the pointer in a 'C' fashion rather than standard Pascal.

Inputs may come from

- keyboard
- smg (screen management)
- fortran file
- pascal file

Default is keyboard KVS 8/89

```

1 5         end
1 4     else
1 4         pascal_file := current_input.ext_file;
1 4         eof_status := eof (pascal_file^);
1 4         if (eof_status) then
1 5 begin
1 5         if (flageof) then
1 5             rstring := ''
1 5         else
1 5             quit ('Aborted on EOF');
1 5             reset (pascal_file^);
1 5         end
1 4     else if (flagerror) then
1 5 begin
1 5         read (pascal_file^, rstring, error := continue);
1 5         error_status := status(pascal_file^);
1 5         readln (pascal_file^);
1 5         if (error_status <> 0) then
1 5             rstring := ' ';
1 5         end
1 4     else
1 4         readln(pascal_file^, rstring);
1 3     end;
1 3 fortran_io:
1 4 begin

```

```

1 4      get_linef (instr, rbuff, current_input.input_unit, flageof,
1 4          flagerror, return_null, eof_status, error_status);
1 4      trim (rstring, rbuff);
1 3  end;
1 3 smg_read, smg_composed_read:
1 4  begin
1 4      rlen := 0;
1 4      if (current_input.input_type = smg_read) then
1 4          error_status := smg$read_string
1 4              (current_input.keyboard, rbuff, instr,
1 4                  received_string_length := rlen,
1 4                  display_id := current_input.display^)
1 4      else
1 4          error_status := smg$read_composed_line
1 4              (current_input.keyboard, current_input.keytable,
1 4                  rbuff, instr, rlen,
1 4                  %immed current_input.display);
1 4      rlen := min(rlen, 12); {truncate to length of rstring}
1 4      if (rlen = 0) then
1 4          rstring := ''
1 4      else
1 4          rstring := substr(rbuff, 1, rlen);
1 4      if error_status = smg$_eof then
1 4          if flageof then
1 5              begin
1 5                  eof_status := true;
1 5                  error_status := 0;
1 5                  rstring := ' ';
1 5              end
1 4          else
1 4              quit ('Aborted on EOF')
1 4          else if error_status = ss$_normal then
1 4              error_status := 0
1 4          else
1 4              if flagerror then
1 4                  rstring := ''
1 4              else
1 4                  lib$stop (error_status);
1 3      end;
1 2      end;
1 1      until (return_null or (length(rstring) > 0));
1 1      if current_input.input_type = terminal_io then
1 1          writeln;
0 0  end;

```

```

1 0 [global] procedure set_input_type
1 0     (itype: input_type_name;
1 0         fortran_unit: integer := 0;
1 0         smg_keyboard: integer := 0; smg_keytable: integer := 0;
1 0         smg_display: integer := 0;
1 0         var pascal_file: [truncate] text);
C 1 0 {
C 1 0     Written December, 1987 by K. Van Stone
C 1 0
C 1 0     Changes the current input type used by get_linep.
C 1 0
C 1 0     Input variables:
C 1 0         itype: input_type_name - The new input type to use. Valid
C 1 0             types are:
C 1 0             terminal_io:      Use the standard terminal input.
C 1 0             fortran_io:       Read from a file specified by a
C 1 0                     fortran unit number.
C 1 0             file_io:        Read from a file specified by a
C 1 0                     pascal file (text) variable.
C 1 0             smg_read:       Use SMG$READ_STRING with the
C 1 0                     specified keyboard.
C 1 0             smg_composed_read: Use SMG$READ_COMPOSED_LINE with
C 1 0                     the specified keyboard and
C 1 0                     keytable.
C 1 0             fortran_unit: integer - The fortran unit number to use when
C 1 0                     itype = fortran_io.
C 1 0             smg_keyboard: integer - The keyboard id to use when itype =
C 1 0                     smg_read or smg_composed_read.
C 1 0             smg_keytable: integer - The key table id to use when itype =
C 1 0                     smg_composed read.
C 1 0             smg_display: integer - The display upon which the keyboard
C 1 0                     echos the text. Not needed if the
C 1 0                     SMG output graphics are not used.
C 1 0             pascal_file: text - The pascal file to read from when
C 1 0                     itype = file_io.
C 1 0
C 1 0     If the input variable is not relevant to itype, then it need not be
C 1 0             specified.
C 1 0 }
1 1 begin
1 1     if (current_input.input_type in [smg_read, smg_composed_read]) and
1 1         (current_input.display <> nil) then
1 1         dispose (current_input.display);
1 1         current_input.input_type := itype;
1 2         case itype of
1 2             fortran_io:
1 2             current_input.input_unit := fortran_unit;
1 2             file_io:
1 2             current_input.ext_file := address(pascal_file);
1
%PASCAL-W-NOTVOLATILE, (1) PASCAL_FILE is non-VOLATILE

```

%PASCAL-I-NOTBEADDR, (1) - may not be parameter to ADDRESS
%PASCAL-W-ADDRESSVAR, (1) PASCAL_FILE is a VAR parameter, ADDRESS is illegal
This compilation warning is related to the one discussed in the
section above. KVS 8/89.

```
1 2      smg_read:  
1 3  begin  
1 3      current_input.keyboard := smg_keyboard;  
1 3  new(current_input.display);  
1 3  if (smg_display <> 0) then  
1 3      current_input.display^ := smg_display  
1 3  else  
1 3      current_input.display := nil;  
1 2  end;  
1 2      smg_composed_read:  
1 3  begin  
1 3      current_input.keyboard := smg_keyboard;  
1 3  current_input.keytable := smg_keytable;  
1 3  new(current_input.display);  
1 3  if (smg_display <> 0) then  
1 3      current_input.display^ := smg_display  
1 3  else  
1 3      current_input.display := nil;  
1 2  end;  
1 1  end;  
0 0  end;
```

-----end-----

Directory.LIS

Directory MTA0:[]

AAAREADME.FIRST;11	ADVANCE.FOR;1	ADVANCE.LIS;1	ADVANCE.OBJ;6
BACKGROUND.FOR;15	BACKGROUND.LIS;1	BACKGROUND.OBJ;10	BDATA.CMN;1
BITOPS.LIS;1	BITOPS.MAR;1	BITOPS.OBJ;4	BLOCK.CMN;2
BYTEOPS.LIS;1	BYTEOPS.MAR;2	BYTEOPS.OBJ;4	CALABIN.FOR;4
CALABIN.LIS;1	CALABIN.OBJ;7	CALESTP.FOR;3	CALESTP.OBJ;1
CALMSTP.FOR;4	CALMSTP.FOR;1	CALMSTP.LIS;3	CALMSTP.LIS;1
CALMSTP.OBJ;9	CALMSTP.OBJ;7	CALNDR.FOR;2	CALNDR.LIS;1
CALNDR.OBJ;8	CH3T4.FOR;3	CH3T4.LIS;1	CH3T4.OBJ;8
CIAOS.LIS;1	CIAOS.OBJ;5	CIAOS.PAS;98	CONVRT.FOR;17
CONVRT.LIS;1	CONVRT.OBJ;8	CTCTS.FOR;2	CTCTS.LIS;1
CTCTS.OBJ;6	DATA.CMN;5	DE.LIS;1	DE.OBJ;3
DE.PAS;2	DE.PEN;3	DEC623.FOR;8	DEC623.LIS;1
DEC623.OBJ;7	DECSTEP.FOR;16	DECSTEP.LIS;1	DECSTEP.OBJ;8
DEDDATA.CMN;4	DETMOD.FOR;16	DETMOD.LIS;1	DETMOD.OBJ;8
DIMISC.INC;1	DIRECTORY.LIS;1	EICS.EXE;10	EICS.EXE;4
EICS.FOR;13	EICS.LIS;1	EICS.MAP;11	EICS.MAP;5
EICS.OBJ;7	FASTPLOT.CMN;7	FILECH.FOR;2	FILECH.LIS;1
FILECH.OBJ;7	FILESPEC.FOR;19	FILESPEC.LIS;1	FILESPEC.OBJ;8
FILLSCI.FOR;10	FILLSCI.LIS;1	FILLSCI.OBJ;9	FOR108.DAT;1
FORMSG.FOR;8	FORMSG.LIS;1	FORMSG.OBJ;5	FPACUM.FOR;20
FPACUM.LIS;1	FPACUM.OBJ;6	FPBUF.CMN;5	FPPRINT.FOR;16
FPPRINT.FOR;15	FPPRINT.LIS;2	FPPRINT.LIS;1	FPPRINT.OBJ;6
FPPRINT.OBJ;5	FP_DATA.DAT;1	GAPCHK.FOR;1	GAPCHK.LIS;1
GAPCHK.OBJ;5	GDE.FOR;7	GDE.LIS;1	GDE.OBJ;4
GETMAG.FOR;6	GETMAG.LIS;1	GETMAG.OBJ;6	GET_FILEDEFF.FOR;8
GET_FILEDEFF.LIS;1	GET_FILEDEFF.OBJ;5	GET_LINEF.FOR;7	GET_LINEF.OBJ;2
INITLZ.FOR;2	INITLZ.LIS;1	INITLZ.OBJ;5	INSEC.LIS;1
INSEC.OBJ;3	INSEC.PAS;2	IOSUBS.LIS;2	IOSUBS.OBJ;14
IOSUBS.PAS;55	IOSUBS.PEN;14	ISIZE.FOR;1	ISIZE.LIS;1
ISIZE.OBJ;6	JMT.FOR;5	JMT.LIS;1	JMT.OBJ;5
KMISC.LIS;2	KMISC.OBJ;11	KMISC.PAS;3	KMISC.PEN;10
KWIKVERT.FOR;10	KWIKVERT.LIS;1	KWIKVERT.OBJ;5	LINK_EICS_PROGRAM.C
NEXTIME.CMN;1	NOOR.FOR;14	NOOR.LIS;1	NOOR.OBJ;4
OADD.CMN;3	OAPAR.CMN;1	PARAM.CMN;6	PASSDATA.FOR;3
PASSDATA.LIS;1	PASSDATA.OBJ;5	PHANGLE.FOR;7	PHANGLE.LIS;1
PHANGLE.OBJ;6	PITCHANG.FOR;1	PITCHANG.LIS;1	PITCHANG.OBJ;5
PITCHANG_360.FOR;6	PITCHANG_360.LIS;1	PITCHANG_360.OBJ;4	PLOT.CMN;3
PUTIN.FOR;35	PUTIN.LIS;1	PUTIN.OBJ;6	QUIT.FOR;3
QUIT.LIS;1	QUIT.OBJ;4	RECOMPILE_OBJECTS.COM;4	
REFRM.FOR;1	REFRM.LIS;1	REFRM.OBJ;5	SBTIME.FOR;8
SBTIME.LIS;1	SBTIME.OBJ;5	SETPARS.FOR;16	SETPARS.LIS;1
SETPARS.OBJ;6	SPIN.OPT;5	SPINBUF.CMN;4	SPINBUF.INC;5
SPINSORT.FOR;21	SPINSORT.LIS;1	SPINSORT.OBJ;8	SPINTROL.CMN;1
SPPLOT.CMN;4	STARTUP.FOR;2	STARTUP.LIS;1	STARTUP.OBJ;5
SYNCSCI.FOR;4	SYNCSCI.LIS;1	SYNCSCI.OBJ;6	SYSMSG.FOR;1

SYSMSG.LIS;1	SYSPAS.OBJ;4	SYSPAS.LIS;1	SYSPAS.OBJ;3
SYSPAS.PAS;40	SYSPAS.PEN;3	TEST.CMD;3	TEST.COM;6
TIMCON.FOR;3	TIMCON.LIS;1	TIMCON.OBJ;5	TRANSBUF.FOR;15
TRANSBUF.LIS;1	TRANSBUF.OBJ;4	UNFLAG.FOR;2	UNFLAG.LIS;1
UNFLAG.OBJ;3	UNPACKOA.FOR;1	UNPACKOA.LIS;1	UNPACKOA.OBJ;4
VAFRD.FOR;25	VAFRD.FOR;20	VAFRD.LIS;3	VAFRD.LIS;1
VAFRD.OBJ;11	VAFRD.OBJ;8	VAXGETC.FOR;2	VAXGETC.LIS;1
VAXGETC.OBJ;4	VAXNUMCK.FOR;2	VAXNUMCK.LIS;1	VAXNUMCK.OBJ;4
VAXWORDEQ.FOR;2	VAXWORDEQ.LIS;1	VAXWORDEQ.OBJ;4	XEDCF.FOR;4
XEDCF.OBJ;1	XES.FOR;9	XES.OBJ;1	XJTF.FOR;3
XJTF.LIS;1	XJTF.OBJ;3	XLS.FOR;1	XLS.LIS;1
XLS.OBJ;3	XMDCF.FOR;6	XMDCF.LIS;1	XMDCF.OBJ;3
XXBKG.FOR;5	XXBKG.LIS;1	XXBKG.OBJ;3	ZEROFILL.LIS;1
ZEROFILL.MAR;2	ZEROFILL.OBJ;3		

Total of 233 files.

DUMP OF FILE SCRATCH: C:\RUNLAND\TICSJ\CS2002A.DAT:1 UN 14-DEC-1990 14:06:19.09
FILE ID (333,59,0) END OF FILE BLOCK 594 / ALLOCATED 594

DEPEN 0001A

01/03/82

VIRTUAL BLOCK NUMBER 1 (00000001), 512 (0200) BYTES

DUMP OF FILE SCRATCH2: BROWNLAND-ECSJC2002A-DAT21 UN 14-DEC-1990 14:06:19.09
FILE ID (335,59,0) END OF FILE BLOCK 594 / ALLOCATED 594

DEA b-0001 A

VIRTUAL BLOCK NUMBER 2 (00000002), \$12 (0280) BYTES